FROM THE AUTHOR OF *THE MYTHICAL MAN-MONTH*

THE DESIGN OF DESIGN

ESSAYS FROM A COMPUTER SCIENTIST

FREDERICK P. BROOKS, JR.

This material is based upon work supported by the National Science Foundation under Grant No. 0608665.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Library of Congress Cataloging-in-Publication Data

Brooks, Frederick P. (Frederick Phillips)
The design of design : essays from a computer scientist / Frederick P. Brooks, Jr.
p. cm.
Includes bibliographical references and indexes.
ISBN 978-0-201-36298-5 (pbk. : alk. paper) 1. Engineering design. 2. Software engineering. 3. Design—Case studies. I. Title.
TA174.B752 2010
620'.0042—dc22

2009045215

ISBN-13: 978-0-201-36298-5
ISBN-10: 0-201-36298-8

Text printed in the United States on recycled paper at Courier in Stoughton, Massachusetts.
First printing, March 2010

Preface

I write to prod designers and design project managers into thinking hard about the *process* of designing things, especially complex systems. The viewpoint is that of an engineer, focused on utility and effectiveness but also on efficiency and elegance.[1]

Who Should Read This Book?

In *The Mythical Man-Month* I aimed at "professional programmers, professional managers, and especially professional managers of programmers." I argued the necessity, difficulty, and methods of achieving conceptual integrity when software is built by teams.

This book widens the scope considerably and adds lessons from 35 more years. Design experiences convince me that there are constants across design processes in a diverse range of design domains. Hence the target readers are:

1. Designers of many kinds. Systematic design excluding intuition yields pedestrian follow-ons and knock-offs; intuitive design without system yields flawed fancies. How to weld intuition and systematic approach? How to grow as a designer? How to function in a design team?

Whereas I aim for relevance to many domains, I expect an audience weighted toward computer software and hardware designers—to whom I am best positioned to speak concretely. Thus some of my examples in these areas will involve technical detail. Others should feel comfortable skipping them.

2. Design project managers. To avoid disaster, the project manager must blend both theory and lessons from hands-on experience as he designs his design process, rather than just replicating

some oversimplified academic model, or jury-rigging a process without reference to either theory or the experience of others.

3. Design researchers. The study of design processes has matured; good, but not all good. Published studies increasingly address narrower and narrower topics, and the large issues are less often discussed. The desire for rigor and for "a science of design" perhaps discourages publication of anything other than scientific studies. I challenge design thinkers and researchers to address again the larger questions, even when social science methodology is of little help. I trust they will also challenge the generality of my observations and the validity of my opinions. I hope to serve their discipline by bringing some of their results to practitioners.

Why Another Book on Design?

Making things is a joy—immensely satisfying. J. R. R. Tolkien suggests that God gave us the gift of subcreation, as a gift, just for our joy.[2] After all, "The cattle on a thousand hills are mine. … If I were hungry, I would not tell *you*."[3] Designing per se is fun.

The design process is not well understood either psychologically or practically. This is not for lack of study. Many designers have reflected on their own processes. One motivation for study is the wide gaps, in every design discipline, between best practice and average practice, and between average practice and semi-competent practice. Much of design cost, often as much as a third, is rework, the correction of mistakes. Mediocre design provably wastes the world's resources, corrupts the environment, affects international competitiveness. Design is important; teaching design is important.

So, it was reasoned, systematizing the design process would raise the level of average practice, and it has. German mechanical engineering designers were apparently the first to undertake this program.[4]

The study of the design process was immensely stimulated by the coming of computers and then of artificial intelligence. The initial hope, long delayed in realization and I think impossible,

was that AI techniques could not only take over much of the drudgery of routine design but even produce brilliant designs lying outside the domains usually explored by humans.[5] A discipline of design studies arose, with dedicated conferences, journals, and many studies.

With so much careful study and systematic treatment already done, why another book?

First, the design process has evolved very rapidly since World War II, and the set of changes has rarely been discussed. Team design is increasingly the norm for complex artifacts. Teams are often geographically dispersed. Designers are increasingly divorced from both use and implementation—typically they no longer can build with their own hands the things they design. All kinds of designs are now captured in computer models instead of drawings. Formal design processes are increasingly taught, and they are often mandated by employers.

Second, much mystery remains. The gaps in our understanding become evident when we try to teach students how to design well. Nigel Cross, a pioneer in design research, traces four stages in the evolution of design process studies:

1. *Prescription* of an ideal design process
2. *Description* of the intrinsic nature of design problems
3. *Observation* of the reality of design activity
4. *Reflection* on the fundamental concepts of design[6]

I have designed in five media across six decades: computer architecture, software, houses, books, and organizations. In each I have had some roles as principal designer and some roles as collaborator in a team.[7] I have long been interested in the design process; my 1956 dissertation was "The analytic design of automatic data processing systems."[8] Perhaps now is the time for mature reflection.

What Kind of Book?

I am struck by how alike these processes have been! The mental processes, the human interactions, the iterations, the constraints,

the labor—all have a great similarity. These essays reflect on what seems to be the underlying invariant process.

Whereas computer architecture and software architecture each have short histories and modest reflections about their design processes, building architecture and mechanical design have long and honorable traditions. In these fields design theories and design theorists abound.

I am a professional designer in those fields that have had only modest reflection, and an amateur designer in some long and deep fields. So I shall attempt to extract some lessons from the older design theories and to apply them to computers and software.

I believe "a science of design" to be an impossible and indeed misleading goal. This liberating skepticism gives license to speak from intuition and experience—including the experience of other designers who have graciously shared their insights with me.[9]

Thus I offer neither a text nor a monograph with a coherent argument, but a few opinionated essays. Even though I have tried to furnish helpful references and notes that explore intriguing side alleys, I recommend that one read each essay through, ignoring the notes and references, and then perhaps go back and explore the byways. So I have sequestered them at the end of each chapter.

Some case studies provide concrete examples to which the essays can refer. These are chosen not because of their importance, but because they sketch some of the experience base from which I conclude and opine. I have favored especially those about the functional design of houses—designers in any medium can relate to them.

I have done functional (detailed floor plan, lighting, electrical, and plumbing) design for three house projects as principal architect. Comparing and contrasting that process with the process of designing complex computer hardware and software has helped me postulate "essentials" of the design process, so I use these as some of my cases, describing those processes in some detail.

In retrospect, many of the case studies have a striking common attribute: *the boldest design decisions, whoever made them, have accounted for a high fraction of the goodness of the outcome*. These bold decisions were made due sometimes to vision, sometimes to

desperation. They were always gambles, requiring extra investment in hopes of getting a much better result.

Acknowledgments

I have borrowed my title from a work of a generation ago by Gordon Glegg, an ingenious mechanical designer, a charming person, and a spellbinding Cambridge lecturer. It was my privilege to lunch with him in 1975 and to catch some of his passion for design. His title perfectly captures what I am attempting, so I reuse it with gratitude and respect.[10]

I appreciate the encouragement of Ivan Sutherland, who in 1997 suggested that I grow a lecture into a book and who more than a decade later sharply critiqued the draft, to its great improvement. My resulting intellectual journey has been very rewarding.

This work has been possible only because of three research leaves granted by UNC-Chapel Hill and my department chairmen, Stephen Weiss and Jan Prins. I was most graciously welcomed by Peter Robinson at Cambridge, Mel Slater at University College London, their department chairmen, and their colleagues.

The NSF Computer and Information Science and Engineering Directorate's Science of Design program, initiated by Assistant Director Peter A. Freeman, provided a most helpful grant for the completion of this book and the preparation of the associated Web site. That funding has enabled me to interview many designers and to concentrate my principal efforts for the past few years on these essays.

I am deeply indebted to the many real designers who have shared their insights with me. An acknowledgments table listing interviewees and referees is an end piece. Several books have been especially informative and influential; I list them in Chapter 28, "Recommended Reading."

My wife, Nancy, co-designer of some of the work herein, has been a constant source of support and encouragement, as have my children, Kenneth P. Brooks, Roger E. Brooks, and Barbara B. La Dine. Roger did an exceptional review of the manuscript, providing dozens of suggestions per chapter, from concepts to commas.

I've been blessed by strong administrative support at UNC from Timothy Quigg, Whitney Vaughan, Darlene Freedman, Audrey Rabelais, and David Lines. Peter Gordon, Publishing Partner at Addison-Wesley, has provided unusual encouragement. Julie Nahil, Full-Service Production Manager at Addison-Wesley, and Barbara Wood, Copy Editor, have provided exceptional professional skills and patience.

John H. Van Vleck, Nobel-laureate physicist, was Dean of Harvard's Division of Engineering and Applied Science when I was a graduate student there, in Aiken's lab. Van Vleck was very concerned that the practice of engineering be put on a firmer scientific basis. He led a vigorous shift of American engineering education away from design toward applied science. The pendulum swung too far; reaction set in; and the teaching of design has been contentious ever since. I am grateful that three of my Harvard teachers never lost sight of the importance of design and taught it: Philippe E. Le Corbeiller, Harry R. Mimno, and Howard H. Aiken, my adviser.

Thanks and praise to The Great Designer, who graciously grants us the means, the daily sustaining, and the joys of subcreation.

Chapel Hill, NC
November 2009

Endnotes

1. The caption for the book cover is based on Smethurst [1967], *The Pictorial History of Salisbury Cathedral,* who adds, "… Salisbury is thus the only English cathedral, except St. Paul's, of which the whole interior structure was built to the design of one man [or one two-person team] and completed without a break."
2. Tolkien [1964], "On Fairy Stories," in *Tree and Leaf*, 54.
3. Psalm 50:10,12. Emphasis added.
4. Pahl and Beitz [1984], in Section 1.2.2, trace this history, starting in 1928. Their own book, *Konstructionslehre*, through seven editions, is perhaps the most important systematization. I distinguish study of the *design process* from rules for design in any particular medium. These are millennia older.

5. The major monograph, tremendously influential, was Herbert Simon's *The Sciences of the Artificial* [1969, 1981, 1996].
6. Cross [1983], *Developments in Design Methodology*, x.
7. A table of the specific design experiences is included in the appendix materials on the Web site: http://www.cs.unc.edu/~brooks/DesignofDesign.
8. Brooks [1956], "The analytic design of automatic data processing systems," PhD dissertation, Harvard University.
9. I thus do not contribute to the design methodologists' goal as stated in http://en.wikipedia.org/wiki/Design_methods (accessed on January 5, 2010):

   > *The challenge is to transform individual experiences, frameworks and perspectives into a shared, understandable, and, most importantly, a transmittable area of knowledge. Victor Margolin states three reasons why this will prove difficult, [one of which is]:*
   >
   > > *'… Individual explorations of design discourse focus too much on individual narratives, leading to personal point-of-view rather than a critical mass of shared values.'*

   To this I must plead, "Guilty as charged."
10. Glegg [1969], *The Design of Design*.

6
Collaboration in Design

A meeting is a refuge from "the dreariness of labor and the loneliness of thought."

BERNARD BARUCH, IN RISEN [1970],
"A THEORY ON MEETINGS"

Menn's Sunniberg Bridge, 1998
Christian Menn, ETH Zürich, ChristianMennPartners AG

Is Collaboration Good Per Se?

Two big changes in design have taken place since 1900:

- Design is now done mostly by teams, rather than individuals.
- Design teams now often collaborate by using telecommunications, rather than by being collocated.

As a consequence of these big shifts, the design community is abuzz with hot topics:

- Telecollaboration
- "Virtual teams" of designers
- "Virtual design studios"

All of these are enabled by telephony, networking, computers, graphic displays, and videoconferencing.

If we are to understand telecollaboration, we must first understand the role of collaboration in modern professional design.

It is generally assumed that collaboration is, in and of itself, a "good thing." "Plays well with others" is high praise from kindergarten onward. "All of us are smarter than any of us." "The more participation in design, the better." Now, these attractive propositions are far from self-evident. I will argue that they surely are not *universally* true.

Most great works of the human mind have been made by one mind, or two working closely. This is true of most of the great engineering feats of the 19th and early 20th centuries. But now, team design has become the modern standard, for good reasons. The danger is the loss of conceptual integrity in the product, a very grave loss indeed. So the challenge is how to achieve conceptual integrity while doing team design, and at the same time to achieve the very real benefits of collaboration.

Team Design as the Modern Standard

Team design is standard for modern products, both those mass-produced and one-offs such as buildings or software. This is indeed a big change since the nineteenth century. We *know the*

names of the leading 18th- and 19th-century engineering designers: Cartwright, Watt, Stephenson, Brunel, Edison, Ford, the Wright Brothers. Consider, on the other hand, the *Nautilus* nuclear submarine (Figure 6-1). We know Rickover as the champion, the Will who made it happen, but which of us can name the chief designer? It is the product of a skilled team.

Consider great designers, and think of their works:

- Homer, Dante, Shakespeare
- Bach, Mozart, Gilbert and Sullivan
- Brunelleschi, Michelangelo
- Leonardo, Rembrandt, Velázquez
- Phidias, Rodin

Most great works have been made by one mind. The exceptions have been made by two minds. And *two* is indeed a magic number for collaborations; marriage was a brilliant invention and has a lot to be said for it.

Figure 6-1 The *Nautilus* nuclear submarine
U.S. Navy Arctic Submarine Laboratory/Wikimedia Commons

Why Has Engineering Design Shifted from Solo to Teams?

Technological Sophistication. The most obvious driver toward team design is the increasing sophistication of every aspect of engineering. Contrast the first iron bridge (Figure 6-2) with its splendid descendant (chapter frontispiece).

The first had to be wrought very conservatively, that is, heavily and wastefully, even though elegantly. Both the properties of the iron and the distribution of static and dynamic stresses were understood imperfectly (though remarkably well!).

Menn's bridge, on the other hand, soars incredibly but confidently, the fruit of years of analysis and modeling.

I am impressed that there are no naive technologies left in modern practice. It was my privilege to tour Unilever's research laboratory at Port Sunlight, Merseyside, UK. I was astonished to find a PhD applied mathematician doing computational fluid dynamics (CFD) on a supercomputer, so as to get the mixing of shampoo right! He explained that the shampoo is a three-layer emulsion of aqueous and oily components, and mixing without tearing is crucial.

Figure 6-2 Pritchard and Darby's Iron Bridge, 1779 (Shropshire, UK)
iStockphoto

The designers of a John Deere cotton-picking machine used CFD to structure the airflow carrying the cotton bolls. A modern farmer spends not only hours on the tractor, but also hours on the computer, matching fertilizer, protective chemicals, seed variety, soil analysis, and crop rotation history.[2] The master cook at Sara Lee adjusts the cake recipe continually to match the chemical properties of the flour coming in; the boss in the paper mill similarly adjusts for the varying pulpwood properties.

Mastering explosive sophistication in any branch of engineering forces specialization. When I went to graduate school in 1953, one could keep up with *all* of computer science. There were two annual conferences and two quarterly journals. My whole intellectual life has been one of throwing passionate subfield interests overboard as they have exploded beyond my ability to follow them: mathematical linguistics, databases, operating systems, scientific computing, software engineering, even computer architecture—my first love. This sort of splintering has happened in all the creative sciences, so the designer of today's state-of-the-art artifact needs help from masters of various crafts.

The explosion in the need for detailed know-how of many technologies has been partially offset by the stunning explosion in the ready availability of such detailed know-how—in documents, in skilled people, in analysis software, and in search engines that find the documents and plausible candidates for collaborators.

Hurry to Market. A second major force driving design to teams is hurry to get a new design, a new product, to market. A rule of thumb is that the first to market a new kind of product can reasonably expect a long-run market share of 40 percent, with the remainder split among multiple smaller competitors. Moreover, the pioneer can harvest a profit bubble while the competition builds up. In the biggest wins, the pioneer continues to dominate. These realities press design schedules hard. Team design becomes a necessity when it can accelerate delivery of a new product in a competitive environment.[3]

Why is this competitive time pressure more intense than before? Global communications and global markets mean that any great idea anywhere propagates more quickly now.

Costs of Collaboration

"Many hands make **light** *work"—Often*
But many hands make **more** *work—Always*

We all know the first adage. And it is true for tasks that are partitionable. The burden on each worker is lighter, hence the time to completion is shorter. But no design tasks are perfectly partitionable, and few are highly partitionable.[4] So collaboration brings extra costs.

Partitioning Cost. Partitioning a design task is itself an added task. The crisp and precise definition of the interfaces between subtasks is a lot of work, slighted at peril. As the design proceeds, the interfaces will need continually to be interpreted, no matter how precisely delineated. There will be gaps. There will be inconsistencies in definition and conflicts in interpretation; these must be reconciled.

To simplify manufacture, there must be standardization of common elements across all the components; some commonality of design style must be established.

And then the separate pieces must be integrated—the ultimate test of interface consistency. It is not just in shipyards where the reality of integration is "Cut to plan; bang to fit."[5]

Learning/Teaching Cost. If n people collaborate on a design, each must come up to speed on the goals, desiderata, constraints, utility function. The group must share a common vision of all of these things—of what is to be designed. To a first approximation, if a one-person design job consists of two parts—learning l and designing d—the total work when the job is shared out n ways is no longer

$$work = l + d$$

but now at least

$$work = n\ l + d$$

Moreover, someone with the vision and knowledge must do the teaching, hence will not be designing. One hopes that the efficiencies of specialization will buy back some of these costs.

Communication Cost during Design. During the design process, the collaborating designers must be sure their pieces will fit together. This requires structured communication among them.

Change Control. A mechanism for change control must be put into place so that each designer makes only those changes that (1) affect only his part or (2) have been negotiated with the designers of the affected parts. Since much of the cost of design is indeed change and rework, the cost of change control is substantial. The cost of *not* having formal change control is much greater.[6]

The Challenge Is Conceptual Integrity!

Much of what we consider elegance in a design is the integrity, the consistency of its concepts. Consider Wren's masterpiece, St. Paul's Cathedral (Figure 6-3).

Figure 6-3 Wren's St. Paul's Cathedral
iStockphoto

Such design coherence in a tool not only delights, it also yields ease of learning and ease of use. The tool does *what one expects* it to do. I argued in *The Mythical Man-Month* that conceptual integrity is *the* most important consideration in system design.[7] Sometimes this virtue is called *coherence*, sometimes *consistency*, sometimes *uniformity of style*. Blaauw and I have elsewhere discussed conceptual integrity at some length, identifying as component principles *orthogonality*, *propriety*, and *generality*.[8] The solo designer or artist usually produces works with this integrity subconsciously; he tends to make each microdecision the same way each time he encounters it (barring strong reasons). If he fails to produce such integrity, we consider the work flawed, not great.

Many great engineering designs are still today principally the work of one mind, or two. Consider Menn's bridges.[9] Consider the computers of Seymour Cray. The genius of his designs flowed from his total personal mastery over the whole design, ranging from architecture to circuits, packaging, and cooling, and his consequent freedom in making trades across all design domains.[10] He took the time to do designs he could master, even though he used and supervised a team. Cray exerted a powerful counterforce against those corporate and external pressures that would have steered his own attention away from design to other matters. He repeatedly took his design team away from the laboratories created by his earlier successes, considering solitude more valuable than interaction. He was proud of having developed the CDC 6600 with a team of 35, "including the janitor."[11]

One sees this pattern—physical isolation, small teams, intense concentration, and leadership by one mind—repeated again and again in the design of truly innovative, as opposed to follow-on, products: for example, the Spitfire team under Joe Mitchell, off at Hursley House, a stately home in Hampshire, UK; Lockheed's Skunk Works under Kelly Johnson, from which the U-2 spy plane and F-117 stealth fighter came; IBM's closed laboratory in Boca Raton, Florida, home of IBM's successful effort to catch up with Apple on the PC.

Dissent

Not everyone agrees with the thesis I have been arguing. Some argue the social justice of participatory design—that it is *right*

for users to have a significant role in the design of objects for their use.[12] Whereas this participation is feasible (and prudent as well as fair) for buildings, user participation in the design of mass-market products is inherently limited to a small sample of prospective users. Such a voice must be conditioned by the representativeness of the sampling, and the vision of the designer.

Others argue that my facts are wrong, that team design has in fact always been the norm.[13] The reader will have to judge for himself.

How to Get Conceptual Integrity with Team Design?

Any product so big, so technically complex, or so urgent as to require the design effort of many minds must nevertheless be conceptually coherent to the single mind of the user.[14] Whereas such coherence is usually a natural consequence of solo design, achieving it in collaborative design is a management feat, requiring a great deal of attention. So, how does one organize design efforts to achieve conceptual integrity?

Modern Design as an Interdisciplinary Negotiation?

Many (mostly academic) writers conclude from the high degree of today's specialization that the nature of design has changed: design today must be done as an "interdisciplinary negotiation" (among the team). The clear implication, though not explicit, is that the team members are peers, and each must be satisfied. *NO!* If conceptual integrity is the final goal, negotiation among peers is the classic recipe for bloated products! The result is design by committee, where none dare say "No" to another's suggestion.[15]

A System Architect

The most important single way to ensure conceptual integrity in a team design is to empower a single system architect. This person must be competent in the relevant technologies, of course. He must be experienced in the sort of system being designed. Most of all, he must have a clear vision of and for the system and must really *care* about its conceptual integrity.

The architect serves during the entire design process as the agent, approver, and advocate for the *user,* as well as for all the other stakeholders. The real user is often not the purchaser. This is evidently true with military acquisitions, where the purchaser (and even the specifier) is far removed from the user. Indeed, the same system may have multiple users, wielding it at strategic, battalion, and personal levels. The purchaser is represented at the design table by marketers. The engineers are represented. The manufacturers are represented. Only the architect represents the users. And, for complex systems as well as for simple residences, it is the architect who must bring professional technology mastery to bear for the users' overall, long-run interest. The role is challenging.[16] I have discussed it in considerable detail in Chapters 4–7 of *The Mythical Man-Month*.

One User-Interface Designer

A major system will require not only a chief architect, but indeed an architectural team. So the conceptual-integrity challenge recurses. Even architecture work must be partitioned, controlled, and hence reintegrated. Here again, conceptual integrity requires special effort.

The user interface, the user's crucial system component, must be tightly controlled by one mind. In some teams, the chief architect can do this detailed work. Consider MacDraw and MacPaint, early Mac tools that were in fact built by their designers. In large architecture teams, the chief architect's scope is too large for him to do the interface himself. Nevertheless, *one* person must do it. If one architect can't master it, one user can't either. At Google, for example, one vice president, Marissa Mayer, maintains personal control over the page format and the home page.[17]

Such an interface designer not only needs lots of using experience and listening skills, he above all needs *taste*. I once asked Kenneth Iverson, Turing Award winner and inventor of the APL programming language, "Why is APL so easy to use?" His answer spoke volumes: "It does what you expect it to do." APL epitomizes consistency, illustrating in detail orthogonality, propriety, and generality. It also epitomizes parsimony, providing many functions with few concepts.

I once was engaged to review the architecture of a very ambitious new computer family, the Future Series (FS) intended by IBM's developers to be a successor to the S/360 family. The architectural team was brilliant, experienced, and inventive. I listened with delight as the grand vision unfolded. So many fine ideas! For an hour, one of the architects explained the powerful addressing and indexing facilities. Another hour, another architect set forth the instruction sequencing, looping, branching capabilities. Another described the rich operations set, including powerful new operators for data structures. Another told of the comprehensive I/O system.

Finally, swamped, I asked, "Can you please let me talk to the architect who understands it all, so I can get an overview?"

"There isn't one. No one person understands it all."

I knew then that the project was doomed—the system would collapse of its own weight. Being handed the 800-page user manual confirmed in my mind the system's fate. How could any user master such a programming interface?[18]

When Collaboration Helps

In some aspects of design the very plurality of designers per se adds value.

Determining Needs and Desiderata from Stakeholders

If deciding *what* to design is the hardest part of the design task, is this a part where collaboration helps? Indeed so! A small team is much better than an individual at studying either an unmet need or an existing system to be replaced. Typically, several minds think of many different questions and kinds of questions. Many questions mean many unexpected answers. The collaborating team must ensure that each member gets full opportunity to explore his trains of inquisitiveness.

Establishing Objectives. Under any design process, the designer begins by conversing with the several stakeholders. These conversations are about the objectives and constraints for the design. The hard task is to flush out the *implicit* objectives and constraints, the ones the stakeholders don't even recognize

that they have. Indeed, from these conversations—what is said, how it is said, what is unsaid—comes the designer's first estimate of the utility function.

A crucial part of this phase is observation of how the user does the job today, with today's tools and circumstances. It often helps to videotape these observations, and to view them over and over.

Having collaborating designers participate is extremely useful for this phase. Extra minds

- Ask different questions
- Pick up different things that are not said
- Have independent and perhaps contradictory opinions of how things are said
- Observe different aspects of working
- Stimulate the discussion of the videotapes

Conceptual Exploration—Radical Alternatives

Early in the design process, designers begin exploring solutions—the earlier the better (as long as no one gets wedded to any solution), for the concreteness of postulated solutions usually elicits hitherto unspoken user desiderata or constraints.

Brainstorming. This is the time for brainstorming. Severally, each member of the design team sketches multiple individual schemes. Collectively, the team members prod each other into radical, even wild, ideas. The standard rules for this stage include "Focus on quantity," "No criticism," "Encourage wild ideas," "Combine and improve ideas," and "Sketch all of them where all can see."[19] More minds mean more ideas. More minds stimulating each other yield lots more ideas.

The ideas are not necessarily better. Dornburg [2007] reports a controlled industrial-scale experiment at Sandia Labs:

> *Individuals perform at least as well as groups in producing quantity of electronic ideas, regardless of brainstorming duration. However, when judged with respect to quality along three dimensions (originality, feasibility, and effectiveness), the individuals significantly ($p<0.05$) out performed the group working together.*

Competition as an Alternative to Collaboration. In the conceptual exploration phase, one can alternatively harness and stimulate the creative powers of multiple designers by holding design competitions. These work best when the known constraints and objectives are concretely stated and shared, and when unnecessary constraints are carefully excised.

In architecture this practice has been routine for centuries. Brunelleschi established himself by winning the design competition for the dome of the Santa Maria del Fiore cathedral in Florence in 1419 (Figure 6-4). His radical concept, its feasibility made plausible by a scale model, opened new vistas, seen today in St. Paul's and the U.S. Capitol.

Figure 6-4 Brunelleschi's Dome, Santa Maria el Fiore
Anonymous, "View of Florence from the Boboli Gardens," 19th Century, Watercolor, Museo di Firenze com'era, Florence, Italy/ Scala/Art Resource, New York.

In architecture and some major civil engineering works, there is a single client and multiple designers hoping to get the job. So a competition naturally suggests itself.

The situation is quite different in the normal product-development environment of a computer or software developer. There it is customary for a single team to be assigned to develop a particular product. There will always be competing ideas inside the team about different design decisions, and debates are routine. But only rarely does a management set up multiple teams to pursue a single objective competitively.

Occasionally, however, there will be a formal design competition within a corporate product-development setting. During System/360 architectural design we worked on a stack architecture for six months. Then came the first cost-estimating cycle. The results showed the approach to be valid for mid-range machines and up, but a poor cost-performer at the low end of the seven-model family.

So we had a design competition. The architecture team self-selected into some 13 little (one- to three-person) teamlets, and each did an architectural sketch, against a fixed set of rules and deadlines. Two of the 13 designs were best in my opinion as judge. They were surprisingly alike, more surprising because the teams were rather cool toward each other and had not communicated.

The confluence of those designs set the pattern for the project. (Their big difference, 6-bit-byte versus 8-bit-byte, occasioned the sharpest, deepest, and longest debate of the whole design process.)

I reckon the design competition, originally suggested by Gene Amdahl, to have been immensely invigorating and fruitful. It put everyone hard to work again after a demoralizing cost estimate. It got each person deeply involved in all aspects of the design, which greatly helped morale and proved valuable in the later design development. It produced a consensus on many design decisions. And it produced a good design.[20]

Unplanned Design Competitions: Product Fights. Not infrequently, it happens that design team B will so evolve its design that it begins to overlap the market objective of design team A. Then one has an ad hoc design competition, a product fight.

I've seen many product fights. They follow a standard script in five acts:

1. The two teams, who may not already know the details of each other's work, meet, compare products and intended markets, and conclude unanimously that there is no real overlap between their products. Both should proceed full speed.
2. Reality appears, in the form of a market forecast or a skeptical boss.
3. Each team changes the design of its product to encompass all of the other product's market, not just the overlapping part.
4. Each team begins wooing supporters among customers, marketing groups, and product forecasters.
5. There comes a shootout before some executive with the power to decide.

Scripts diverge at this point: team A wins; team B wins; both survive; neither survives the intense scrutiny engendered by the competition.

This scenario can and usually should be shortened by early action by a skeptical boss. Sometimes, however, it may be the best way to get a thorough (and impassioned) exploration of two quite different design approaches.

Design Review

The phase of design where collaboration is most valuable, even necessary, is design review. Multiple disciplines must review: other designers, users and/or surrogates, implementers, purchasers, manufacturers, maintainers, reliability experts, safety and environment watchdogs.

Each disciplinary specialist must review the design documents alone, for careful review takes time, reflection, and perhaps the study of references, archives, and other designs.[21] Each will bring a unique point of view; each will raise different issues and find different flaws. But joint, group review is also imperative.

Demand Multidisciplinary Group Review. Group review has the power of numbers, but special power comes from the viewpoints of multiple disciplines. The review team should be much

larger than the design team. Those who will build the design, those who will maintain it, sample users, those who will market it—all must be included. Consider the review for a new submarine design. The supply officer sees a shortcoming; his spoken concern triggers a similar concern for the damage control specialist. The manufacturing tooling expert sees something hard to build; his suggested solution sets off alarms in the acoustic expert's mind.

Designers at the Electric Boat Division of General Dynamics told me of a review in which the shipyard foreman took one look at a semicylindrical storage tank and quickly suggested rolling a one-piece cylinder, cutting it in half, and roofing it with a flat plate. This was in place of some 20 pieces the engineer had specified. Said the foreman, "We submarine builders are good at rolling cylinders."

Similarly, a designer at Brown & Root in Leatherhead, England, told me of a design review for a deep-sea oil-drilling platform. The maintenance foreman pointed to a particular unit and said, "Better make that one out of heavy-gauge steel."

"Why?"

"Well, we can paint it in the workshop before it's installed, but where it goes, we'll never be able to paint it again."

The engineers redesigned the whole vicinity of the platform so the unit could be reached.

Use Graphical Representations. For design review, the most important aid is a common model of the product—a drawing, a full-scale wooden mock-up or virtual-reality simulation of a submarine, a prototype of a mechanical part, perhaps an architectural diagram of a computer.

A multidisciplinary design review often demands a richer variety of graphical representations of the design than the designers themselves have been using. Not everyone in the review will be able to visualize the end product from the engineering/architectural drawings. My observation from visiting various facilities is that such design reviews are probably the most fruitful applications of virtual-environment visualization technology.[22]

Sharing the product model and sharing each other's comments are both vital to effective design review; tools for simulating such sharing are the sine qua non of group design reviews where all the players cannot be physically present. Here telecollaboration comes into its own.

When Collaboration Doesn't Work—for Design Itself

The Fantasy Concept of Design Collaboration. The computer-supported-collaborative-work literature is peppered with a fantasy version of collaborative design. This would be harmless, except that the fallacious concept focuses ever more elaborate academic research on ever less useful technological tools for collaboration.

In this fantasy, a design team really or virtually sees a model of the design object—whether a house, a mechanical part, a submarine, a whiteboard diagram of software, or a shared text. Any team member proposes changes, usually by effecting the change directly in the model. Others propose amendments, discussion proceeds, and bit by bit the design takes form.

Not How Collaborators Design. But the fantasy concept doesn't fit how collaborators really do design, as opposed to design review.

In all the multi-person design teams I've seen, each part of a design has at any time one owner. That one person works alone preparing a proposal for the design of his part. Then he meets with his collaborators for what is in effect a micro-session of design review. Then he normally retires and works out the detailed consequences of the decisions and directions discussed collaboratively.

If alternate proposals are made in the session, and not accepted by the owner, the proposer will often withdraw and develop an alternate design. Then the session will convene again, to choose, fuse, or strike off in some third direction.

Where's Design Control? The fantasy concept has no function for originating designs, only refining them. The fantasy concept is

flawed as a model for collaborative design change, too. Schedule gain from collaboration implies concurrent activity; and concurrent activity requires *synchronization*, a step totally missing from solo design. Designer Jack owns the air ducts in an oceangoing tanker; Jill owns the steam pipes. As each fleshes out his design, and at every subsequent change, some mechanism of *design control* must monitor that they don't both use the same space. Some *resolution procedure* must be in place for settling conflicts. Some *version control* must be established so that each designs against a *single time-stamped version* of all the earlier design work.

In one instance of the fantasy concept I have actually seen proposed, the client admiral views the design model for a nuclear submarine, and he moves a bulkhead to give equipment repairers better access. (Making this possible is a technically challenging task in a virtual-reality interface to a CAD system. Many techniques for real-time visualization depend upon the static nature of most of the world-model.)

But the challenge is not worth accepting! The admiral may want to move the bulkhead to see how the space will look and feel, and he may be allowed to do that in a playpen version of the model. But before any such move becomes part of the standard design version, someone or some program must check the effects on the space on the other side of the bulkhead, the structural consequences, the acoustic consequences, the effects on piping and wiring. Imagine the horror of the responsible engineers to find that the bulkhead has been moved by the admiral, who cannot possibly have known the constraints and design compromises it embodied. By the time there is a design for the admiral to walk through virtually, it is far enough along to require formal change control.

The fantasy model of collaborative design reflects a monumental unconcern about conceptual integrity. Jill pats the design here; Jim nudges it there; Jack patches it yonder. It is spontaneous; it is collaborative; and it produces poor designs. Indeed, we know the process so well that we have a scornful name for it—*committee design*. If collaboration tools are designed so they encourage committee design, they will do more harm than good.

Conceptual Design, Especially, Must Not Be Collaborative

Once the exploratory stage is past and a basic theme is selected, it's time for conceptual integrity to rule. A design flows from a chief designer, *supported by* a design team, not *partitioned among* one.[23]

To be sure, the conceptual design thus pursued may run into a blind alley. Then a different basic scheme must be selected, and collaborative exploration is again in order until that new basic scheme is selected.

Two-Person Teams Are Magical

The foregoing discussion of design collaboration dealt with teams of more than two people. Two-person teams are a special case. Even in the conceptual design stage, when conceptual integrity is most imperiled, pairs of designers acting *uno animo* can be more fruitful than solo designers. The literature on pair programming shows this to be true during detailed design. Typical initial productivity runs less than two working separately, but error rates are radically reduced.[24] Since perhaps 40 percent of the effort on many designs is rework, net productivity is higher and products are more robust.

The world is full of two-person jobs. The carpenter needs someone to hold the other end of the beam. The electrician needs help when feeding wire through studs. Child raising is best done by two actively collaborating parents. "It is not good for man to be alone," while spoken in its truest sense about marriage,[25] might usefully be preached to lone-ranger designers.

The typical dynamics of two-person design collaboration seem different from those of multi-person design and solo design. Two people will interchange ideas rapidly and informally, with neither a protocol as to who has the floor nor domination by one partner. Each holds the floor for short bursts. The process switches rapidly among micro-sessions of proposal, review and critique, counterproposal, synthesis, and resolution. There is typically a *single thread* of idea development, without the maintenance of

separate individual threads of thought as in multi-person discussions. Two pencils may move over the same paper with neither collision nor contradiction.

"As iron sharpens iron," each stimulates the other to more active thought than might occur in solo design. Perhaps the very need to articulate one's thinking—to state *why* as well as *what*—causes quicker perception of one's own fallacies and quicker recognition of other viable design alternatives.

A classic 1970 paper by Torrance showed that dyadic interaction produced twice as many original ideas, produced ideas of twice as much originality, increased enjoyment, and led subjects to attempt more difficult tasks.[26]

Pair-wise design sessions still need to be interspersed with solo ones—to detail, to document the creative fruit, and to prepare proposals for the next joint session.

So What, for Computer Scientists?

Much effort by academic computer scientists has gone into the design of tools for computer-assisted collaboration by workers in their own and other disciplines. Distressingly few of these ideas and tools have made it into everyday use. (Important tools that have succeeded are code control systems and "Track Changes" in Word.) Perhaps this is because it is especially easy for academic tool builders to overlook some crucial properties of real-world team design:

- Real design is always more complex than we tend to imagine.[27] This is especially true since we often start with textbook examples, which have perforce been oversimplified. Real design has more complex goals, more complex constraints to be satisfied, more complex measures of goodness to be satisficed. Real design always explodes into countless details.
- Real team design always requires a design-change control process, lest the left hand corrupt what the right hand has wrought.
- No amount of collaboration eliminates the need for the "dreariness of labor and the loneliness of thought."

For these reasons, I think we should be very leery about assigning graduate students with little or no real-world design experience dissertation topics in the field of collaborative design tools. Moreover, our journals should be very slow to accept such papers that are not based on real-world experience and/or real design applications.

Notes and References

1. This marvelous phrase was quoted by Bernard Baruch, who said his attorney said it to him.
2. *Economist* [2009], "Harvest moon."
3. The wise manager of a multi-project organization early launches a solo designer, or a pair, to start exploring designing for a technology foreseeable, but not yet buildable.
4. Brooks [1995], *The Mythical Man-Month,* Chapter 2.
5. Shipyard foreman at Electric Boat, Groton, CT (personal communication).
6. The most complete scientific study I have seen comparing solo and individual designers is Cross [1996a], *Analysing Design Activity*.

> *The Delft protocols included a solo designer and a three-person team attacking the same problem, with both observed by video and the solo designer encouraged to think aloud. Twenty different chapters, each using its own analytical method, analyze the Delft video protocols. Most apply their authors' own predefined categories of activity to one or both of the protocols. Many of the chapters either compare the activities and performance of the two alternatives or else analyze the social behavior of the team. The most specific conclusion is that by Gabriela Goldschmidt [1995], "The designer as a team of one": "Detailed analysis leads to the conclusion that there are almost no differences between the individual and the team in the way they bring their work to fruition."*

Charles Eastman [1997] reviews this book in *Design Studies* (475–476):

> *Together the studies offer a rich set of perspectives that allow a reader to understand both the fertileness and the idiosyncrasy of design processes. The video transcription obviously captured a rich characterization of design behavior, . . . The limitation of current methods of protocol analysis, however, are made readily apparent. Each study by itself provides only a small peephole into the overall design process. Only through the cumulative breadth of multiple studies does the sense of the full process emerge.*
>
> *This book clearly presents the current state of design protocol studies after thirty years of effort and relates them more generally to various theories of design.*

7. Brooks [1995], *The Mythical Man-Month*, Chapter 4, 42ff.
8. Blaauw and Brooks [1997], *Computer Architecture*, Section 1.4; Brooks [1995], *The Mythical Man-Month*, Chapters 4–7, 19.
9. Billington [2003], *The Art of Structural Design*, Chapter 6; Menn [1996], "The Place of Aesthetics in Bridge Design."
10. Blaauw and Brooks [1997], *Computer Architecture*, Chapter 14.
11. Murray [1997], *The Supermen*.
12. Greenbaum and Kyng [1991], *Design at Work*; Bødker [1987], "A Utopian Experience."
13. Weisberg [1986], *Creativity: Genius and Other Myths*; Stillinger [1991], *Multiple Authorship and the Myth of Solitary Genius.*
14. R. Joseph Mitchell, the designer of the Spitfire, warned one of his test pilots (the user!) about engineers: "If anybody ever tells you anything about an aeroplane which is so bloody complicated you can't understand it, take it from me: it's all balls."
15. Eoin Woods of Artechra says,

> *I'm not as pessimistic as you about joint design. I've worked in teams where we had spirited discussion to drive our designs and then agreed the solution among us (albeit sometimes with a benign dictator making final decisions). The designs remained coherent because it was one or two strong concepts of the design that won out and then drove all of the other decisions; we didn't design by committee and "horse-trade" the detailed decisions (personal communication [2009]).*

16. Brad Parkinson, now at Stanford, one of the two system architects/contracting officers for the GPS system, pointed out that the challenges of that task were substantially increased by having multiple contractors for the several system pieces (personal communication [2007]).
17. Holson [2009], "Putting a bolder face on Google."
18. Mary Shaw of Carnegie-Mellon asks, "What does this say about modern software development environments and their APIs?"
19. Osborn [1963], *Applied Imagination*.
20. Design competitions in organization design are yet different; the task is inherently political. The various competing forces usually do not even share the objective of getting the organization that works best. How well the organization will work is subordinated to who will have which levers of power.
21. Margaret Thatcher: "One wants documents [as opposed to viewgraph foils] so one can think through beforehand, and consult colleagues" (personal communication via Sir John Fairclough). American business all too often does reviews via PowerPoint presentations. Those vague bullets enable each participant to interpret the information as he pleases; they also facilitate the suppression of embarrassing but crucial details.

    Lou Gerstner, turnaround CEO of IBM, startled the whole culture early on ([2002], *Who Says Elephants Can't Dance?*, 43): "Nick was on his second foil when I stepped to the table and as politely as I could in front of his team, switched off the projector . . . it had a terribly powerful ripple effect . . . Talk about consternation. It was as if the President of the United States had banned the use of English at White House meetings."
22. Brooks [1999], "What's real about virtual reality?"
23. Harlan Mills's concept of a supported-chief-designer team, a "surgical" team, is detailed in Brooks [1995], *The Mythical Man-Month*, Chapter 3.
24. Williams [2000], "Strengthening the case for pair-programming"; Cockburn [2001], "The costs and benefits of pair programming."
25. Genesis 2:18.
26. Torrance [1970], "Dyadic interaction as a facilitator of gifted performance."

27. See, for example, the impressive PhD dissertations by Hales [1991], "An analysis of the engineering design process in an industrial context" (Cambridge), or Salton [1958], "An automatic data processing system for public utility revenue accounting" (Harvard), for detailed documentation of what is involved in an actual design.

Subject Index

Note: Boldface page references indicate definitions or the beginning point of a substantial treatment. Page numbers in those scopes are not separately listed under that term.